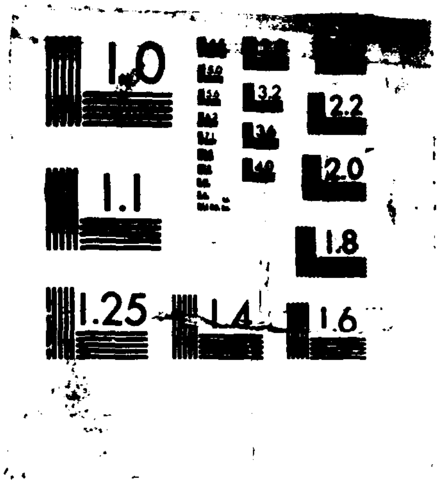
AD-A182 357 TEMPLATE-SET APPROACH TO VLSI (VERY LARGE SCALE INTEGRATED) PATTERN INSPECTION(U) STANFORD UNIV CA CENTER FOR INTEGRATED SYSTEMS S CHAE ET AL. 1984 1/1

UNCLASSIFIED MDA903-84-K-0062 F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART

Template-Set Approach to VLSI Pattern Inspection

Soo-Ik Chae, James T. Walker,
David H. Dameron, Chong-Cheng Fu, James D. Meindl
Center for Integrated Systems, Stanford University, Stanford, CA 94305

1984

Abstract

A new approach is described for the automatic detection of defects in VLSI circuit patterns such as photomasks and wafers. It is based on morphological feature extraction using templates that represent a set of local pixel configurations within a specified window. These templates are stored in content-addressable memories (CAMs) to facilitate parallel comparisons of window-pattern scanning over a tested image. Maskable CAMs reduce the size of a template set substantially. Two error-detection algorithms are implemented to detect both random defects and dimensional errors.

Introduction

As the minimum feature size of VLSI patterns is reduced and the number of mask levels is increased, the inspection of masks and wafers becomes increasingly important to maintain a high yield. The inspection methods based on binary image patterns have been proposed[1] and are classified as *reference methods* for pixel-by-pixels comparisons and *local-property methods* based on local-pattern configurations. The local-property method does not require computer aided design (CAD) data as a reference or rigid alignment because every decision is based only on a local pattern; however, some patterns that do not violate the design rules, such as missing or additional patterns, cannot be detected. These errors are rarely caused by the patterning process itself and can be eliminated by reliable CAD and pattern-generation tools.

The VLSI patterns on the masks and wafers are normally restricted. Usually their edges are straight, and their directions are locked on a small set of angles such as {0, 90, 180, 270} in the Manhattan-style layout. In addition, there are design rules that limit the degrees of freedom of the patterns. Although acquisition of the pattern is another difficult problem, a binary image of the circuit patterns is assumed to be available for real-time inspection. This paper focuses only on image analysis.

An objective of pattern inspection is to locate the center of every window that contains an unacceptable pattern. This can be achieved by comparing a windowed image to a set of templates that represent the morphological properties of the circuit patterns. This template-set approach to pattern inspection of a binary image consists of the following steps:

Step 1: collect all acceptable or unacceptable local patterns as a template set

Step 2: obtain a local windowed image centered at a pixel

Step 3: determine whether any template matches the window pattern

Step 4: repeat steps 2 and 3 until all the pixels in the inspected image are scanned

Because the patterns are quite restricted, the acceptable or unacceptable patterns are well-defined. All possible acceptable patterns or their complement must be stored as a template set. There are several difficulties in generating the template set,[2] however, and a suitable approach must satisfy several conditions.

- ***Reasonable size of the template set.*** The template set is stored as a look-up table in a memory and, as a result, the number of templates must be reasonably small because the size of the memory is limited so as to be manageable.
- ***Completeness of the template set.*** A template set for an algorithm must include all templates that match the window patterns determined by the algorithm.
- ***Fast searching.*** Inspection throughput is determined by the speed of searching for any template matched to the window pattern.
- ***High defect coverage.*** The two types of defects that must be detected in VLSI pattern inspection are random and dimensional errors. The defect coverage of an inspection algorithm depends on the size of the window. It is difficult to detect both types of errors with a reasonably small window.

Goto[3] and Jarvis[4] published papers describing this template-set approach. Goto used a 3 × 3 window to lessen the template number. Each window pattern was classified into one of 16 predetermined categories and represented by a 4-bit word. Because the window was small, its capability of defect detection was poor. As a result, a hierarchical method based on a 5 × 5 array of 4-bit words was employed to detect a defect larger than the window. Jarvis applied this approach as a filtering step because it was difficult to obtain a small complete template set. Although the template set was not complete, it could filter out most of the acceptable window patterns so that the number of patterns to be further analyzed was minimized.

Solution

A solution to the above problems is proposed for easy implementation in VLSI circuits. An objective of the inspection is to detect unacceptable deviations from the intended patterns. As a measure of deviation, *edge roughness* must be defined so that random error can be determined precisely, and this will require additional definitions. For a discrete image I in a square grid system, I(i,j) is a square picture element (pixel) where (i,j) represents its position. A pixel with the value of 0 (1) is called a 0 (1). A *4-neighbor* of a pixel I(i,j) is any pixel I(m,n) such that $|m-i| + |n-j| = 1$. An *8-neighbor* of a pixel I(i,j) is any I(m,n) such that $\max\{|m-i|, |n-j|\} = 1$. A *connected* pattern is a set of 1s such that each has at least a 4-neighbor equal to 1. An *edge pixel* is any 1 with at least a 4-neighbor equal to 0. A *contraction* operation C to a window pattern P sets every pixel p to 0 such that p is 1 and has at least an 8-neighbor in the window equal to 0. The pattern resulting from the application of k contraction operations to the window pattern P is denoted by C^kP. For window patterns A and B, $A \subset B$ if $B(i,j) = 1$ for all i,j such that $A(i,j) = 1$ and at least one pixel position exists such that $B(i,j) = 1$ and $A(i,j) = 0$. A *noise-free window pattern* has a straight edge with zero or one corner in the window. A pattern with an *edge roughness* of k is defined as any P such that P is a connected pattern, and a noise-free window pattern R exists such that $C^kR \subset P \subset R$, $P \not\subset CR$ and $C^{k-1}R \not\subset P$. It must be noted that this edge roughness is not a global but a local concept that is applied to window patterns. Pixel size is assumed to be small enough that patterns with an edge roughness of 1 are acceptable. Random-error patterns are thereby defined as patterns with an edge roughness of more than 1. This solution has three significant advantages.

- ***Bit maskability.*** The number of templates can be reduced substantially by changing every edge pixel of the template pattern into "don't-care." The number of required templates with "don't-cares" increases polynomially as the window size increases linearly, but it rises exponentially without "don't-cares." The completeness of the template set can be achieved easily because its size is reasonably small. In addition, the false error detection rate is reduced. As a result of quantization error, some degree of edge roughness may occur in the discrete binary image; however all the patterns with an edge roughness of 1 must be accepted. False error detection can be avoided by masking all pixel positions corresponding to edge pixels of the intended patterns during the comparison operation. This is an effective approach to the quantization-error problem.
- ***Content addressable memory.*** The speed of the matching operation can be independent of the size of the template set if the templates are stored in a content-addressable memory. Bit masking can also be implemented for "don't-care" conditions.
- ***Multiple algorithms.*** Two algorithms for *random error detection* (RED) and *dimensional error detection* (DED) are used to enhance the defect coverage in VLSI pattern inspection. There is no speed penalty because both algorithms can be processed in parallel.

This solution can be implemented in hardware for real-time inspection at the video signal rate because of its high speed and parallelism. The following steps are required to apply this approach to a specific problem.

- Determine the type of acceptable patterns.
- Develop error-detection algorithms and select a window for each.
- Construct a template set for each algorithm.

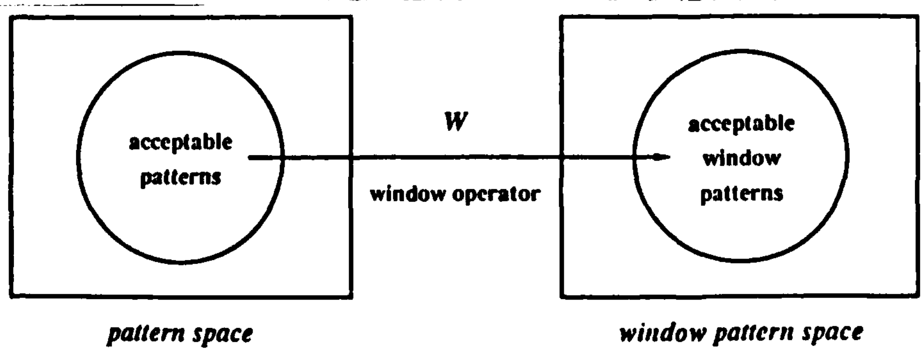


Figure 1. Mapping of patterns to the window-pattern space.

The window-pattern space is an N-dimensional binary space $\{0,1\}^N$ where N is the number of bits in the window. After the acceptable patterns are defined, the inspection algorithm to be used can be determined. A window for this algorithm is then chosen, and patterns are mapped into the window-pattern space as illustrated in Figure 1. There are two sets in the window-pattern space – one for acceptable and the other for unacceptable window patterns; either one can be selected as a template set.

Error-Detection Algorithms

In VLSI pattern inspection, the two types of defects of interest are random defects and dimensional errors. High defect coverage can be obtained by two detection algorithms; one is a RED algorithm that can detect irregular shapes and the other is a DED algorithm that can detect patterns with widths or gaps smaller than their minimum allowable values.

A RED algorithm is represented by the statement, *random-defect patterns have an edge roughness of greater than 1*. More precisely, however, some of the unacceptable patterns are characterized by edges that are not oriented in any allowable direction. In this algorithm, a square window is suitable for the extraction of the edge profile of the patterns. It should be large enough to detect most random features but must contain only one corner or less of the acceptable pattern. If two corners are permitted, the number of templates in the set will increase by approximately an order of magnitude.

All acceptable window patterns are included in the RED template set because the subset in the window space corresponding to the acceptable patterns is smaller and easier to generate. The layout rules must be restricted so that the template set is simpler to generate. Assume that the layout is based on the Mead-Conway design rules[5] and all corners are on the λ grid points. It is also assumed that the minimum feature size equal to 2 λ is 10 pixels. The maximum size of the window is then limited to 5 pixels in each side so that the window permits only one corner.

The template patterns for the RED algorithm are local *two dimensional* (2-D) patterns because its defect coverage is superior to that obtained from any other local-property method that does not fully use the 2-D information. The RED algorithm for the 1 λ square window(Figure 2a) can be refined by the corner number in the window, edge roughness, and edge direction and can be stated as follows.

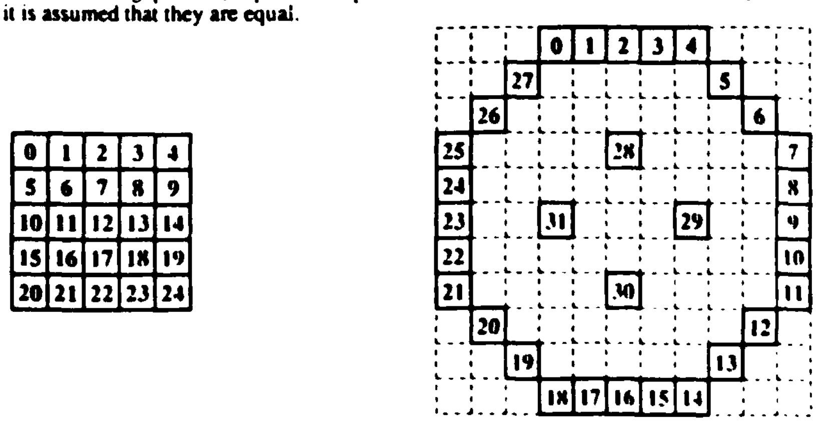
If there is one corner or less in the window
and the direction of any edge is one of the locked directions
and the roughness of any edge is less than or equal to 1,
the window pattern is acceptable.

A pattern can be acceptable in the RED window but unacceptable in the DED window(Figure 2b) because the minimum gap and width requirement is violated. The DED algorithm is as follows.

There is an error pattern in the window if
(1) there is a pair of parallel edges within a circle or if
(2) some pixels within the circle differ in value from those on the circle
when all pixels on the circle have the same value.
Here, the diameter of the circle is equal to the minimum feature size.

Because a square grid system is employed, an octagon must be used as an approximated circle. All pixels inside the octagon need not be in the window because of redundancy. All pixels in the boundary of the octagon (condition 1) and several additional pixels inside the octagon (condition 2) must be included in the window, as illustrated in Figure 2b. The unacceptable patterns are selected as the template set for the DED algorithm because it can be represented by a smaller number of templates.

If the minimum allowable width and gap differ, separate template sets and windows are needed for gap and width error detections, respectively; for simplicity, it is assumed that they are equal.



(a) (b)

Figure 2. Window shapes for DED and RED. (a) A 25-bit RED square window. (b) A 32-bit octagon DED window.

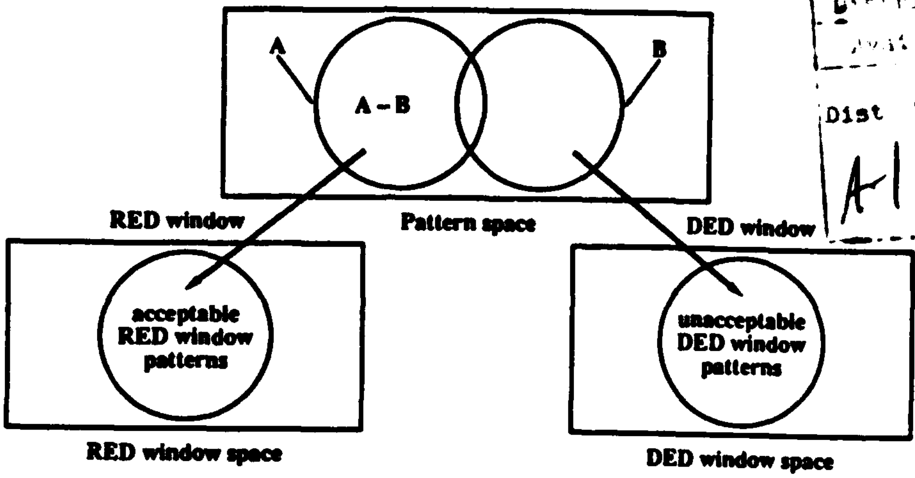


Figure 3. Acceptable pattern sets for the RED/DED algorithms.

In Figure 3, the subsets of patterns whose window patterns match the template sets from RED and DED are labeled A and B, respectively. The set of acceptable patterns is $A - B$ because the RED template is for acceptable window patterns and the DED template is for window patterns with errors. The DED algorithm can be adjusted to eliminate all unacceptable patterns from the subset corresponding to the RED template set.

Template Set

For a window w with N bits, a *template* is defined as a subset of the window space $\{0,1\}^N$ and is represented as a doublet with two N-bit words $\{ t_p, t_m \}$ – a *pattern word* t_p and a *mask word* t_m. If $w_k = t_{pk}$ for all k such that $t_{mk} = 0$, w is *matched* to the template $\{t_p, t_m\}$, where $t_p = (t_{p1}, t_{p2}, \ldots, t_{pN})$, $t_m = (t_{m1}, t_{m2}, \ldots, t_{pN})$, $w = (w_1, w_2, \ldots, w_N)$. If $t_{mi} = 1$, then t_{pi} is "don't-care" and w_i will be matched to t_{pi}. Actually, a template can be implemented with a maskable CAM. If w is matched to at least one template in the template set T, T is said to match w. If T matches only every acceptable window pattern, it is complete for acceptable patterns; if it matches at least one window pattern during the scanning only for each error pattern, it is complete for unacceptable patterns.

Template-Set Generation

A template set is optimal when it is small in size and maximum in defect coverage. The template set can be generated by the empirical collection method, primitive pattern method, and algorithmic property method.

- ***Empirical collection method.*** Window patterns can be collected as templates by scanning a defect-free image until there are no more templates. It is an empirical extraction process of local patterns without direct use of an error-detection algorithm; however, it is simple and straightforward. This method encounters two problems. First, it is difficult to ensure the completeness of the template set because it is not easy to scan all possible patterns and, second, it is difficult to determine when to finish collecting the templates.
- ***Primitive pattern method.*** Because VLSI circuit patterns can be represented by a set of primitive patterns, real images are

pixel number	0	24
pattern word	0011100111001111111111111	
mask word	0010000100001001100000000	

(a)

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

(b)

Figure 4. Examples of a RED template and a window pattern. (a) RED template. (b) Window pattern matched to it.

not necessary to generate the template set. In the Manhattan style, the primitive patterns are a square pattern with each side larger than the minimum feature size and its complementary pattern, which is a square hole. It is basically the same as the empirical method but extraction from only primitive patterns is required to guarantee the completeness of the template set.

- *Algorithmic property method.* Template generation can be simplified by using the local property of the patterns in the image and the property of the algorithm itself. It is an effective method if the patterns required to match templates are simple in the error detection algorithm.

RED Template Set

It is easiest to construct a RED template set for acceptable patterns by using the primitive pattern method because the primitives are small in number. A template-generation algorithm, as outlined below, obtains window patterns from an image that includes only primitive patterns and changes each of its edge pixels to "don't-care." It guarantees the completeness of the template set.

```
begin(* template set generation *)

    set templateset empty;
    repeat
        read a new pixel data;
        form a new window pattern W;
        T = W;
        for each pixel W(i,j) do begin
            if W(i,j) is an edge pixel then begin
                T(i,j) = don't care;
            end else begin
                T(i,j) = W(i,j);
            end;
        end;
        if T is not in templateset then begin
            templateset = templateset + T;
        end;
    until all pixel data are scanned;

end;(* template set generation *)
```

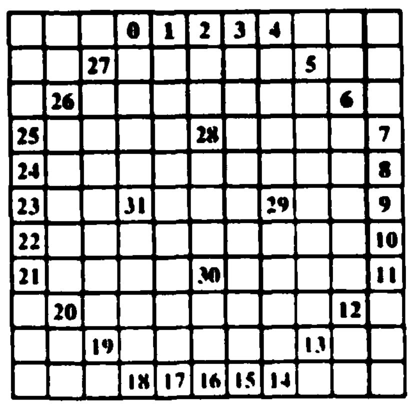
As an example, a window pattern and its matched templates are illustrated in Figure 4.

DED Template Set

The DED template set is constructed only for unacceptable patterns and its purpose is to detect the errors that cannot be distinguished by the RED algorithm. All irregular patterns can be detected by the RED algorithm; however, unacceptable rectangular patterns that violate the minimum width and gap requirement can also be matched to a RED template. These errors can be observed by creating a DED template that corresponds to every possible rectangular error pattern positioned at the center of the DED window such that it must be matched to the template at least once during scanning. The algorithmic property method is suitable for template-set generation because rectangular patterns are simple. An error pattern and a template matched to it are illustrated in Figure 5.

pixel number	0	31
pattern word	00000000000000000000000000001011	
mask word	11111111000111111111110001110100	

(a)



(b)

Figure 5. Examples of a DED template and an error pattern. (a) DED template (b) Error pattern matched to it

Analysis and Simulation of Defect Coverage

Defect coverage is perfect if there is neither a missed defect nor a false detection. The probability of false detection caused by noise is assumed to be negligible. To evaluate defect coverage with two RED and DED templates sets, it can be verified by simulation but coverage analysis is more effective. It is assumed that the RED square window is N pixels in each side and the DED octagon window

is $2N + 1$ in diameter where the minimum feature size is $2N$ pixels. The following type of defects can be completely detected:

- **Every pattern with an edge roughness of 2 or more.** All patterns except those with an edge roughness of 1 or 0 can be detected as errors by the RED algorithm because a RED template set includes all possible templates for patterns with an edge roughness of 1 or 0.
- **Every rectangular pattern with a minimum width or gap of less than N-1 pixels.** It can be detected by the RED template set because it has two parallel edges in the window; however, there is no template for the window pattern with two parallel edges in the RED template set.
- **Every rectangular pattern with a minimum width or gap of less than $2N$ pixels and greater than N-2 pixels.** It can be detected by the DED template set. If its maximum width or gap is greater than or equal to $2N$, it is detected by condition (1) of the DED algorithm; otherwise, it is detected by condition (2).

In summary, every pattern with a random shape and every rectangular pattern with a width or gap of less than $2N$ pixels can be detected by the RED and DED template sets. Undesired additional rectangular patterns whose minimum widths are greater than $2N$ cannot be detected. The probability of such error patterns, however, is negligible.

The algorithms have been simulated to verify their correctness and the defect coverage, and the results were also used to determine the number of detections for a particular error. Figure 6 is a binary SEM image of contact-hole patterns in photoresist. The RED templates were generated to tolerate the rounding effect at each corner, and the DED templates were generated to detect only feature sizes that were less than 80 percent of the allowable minimum size. These constraints resulted in an RED set with 128 templates and a DED set with 16 templates. A simulation result with these template sets is shown in Figure 6. The positions where error patterns exist are marked with a star for dimensional error and a small circle for random error. For convenience the inspection operation was not applied in the region outside of dotted line. The two types of errors in this image were the unintended bridging and narrow gap between contact holes caused by overdevelopment of resist, and both were correctly detected.

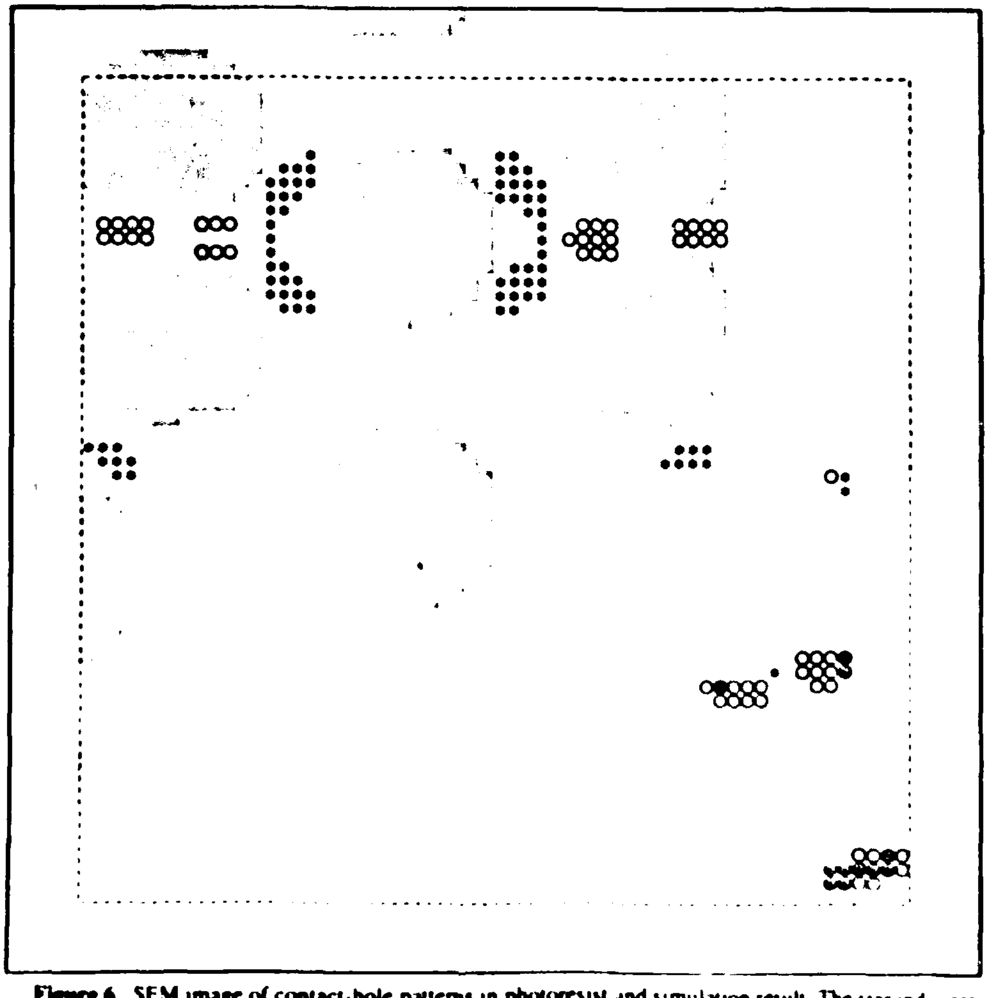


Figure 6. SEM image of contact-hole patterns in photoresist and simulation result. The star indicates DED, and the small circle denotes RED. Inspection is performed just inside the dotted box.

The results of RED template generation for various sizes of square windows in the Manhattan-style patterns are summarized in Table 1. As the window increases, the memory required to store the templates grows rapidly,which limits the size of the window.

Table 1: Number of RED templates versus size of window.

Window dimension N	3 × 3	4 × 4	5 × 5	6 × 6	7 × 7
Window size (bits) N^2	9	16	25	36	49
Template size (bits) $2N^2$	18	32	50	72	98
Template number T	32	72	128	200	288
Memory required (bits) M	576	2304	6400	14400	29264

Custom VLSI Circuits

As illustrated in Figure 7, two functional blocks for windowing and comparing are required to inspect a serial binary input as a tested image. The windowing block forms a 2-D window pattern from a serial video signal and is basically a set of shift registers. Its output is connected to the bit-line drivers that drive the bit lines of the CAM array of the comparing block.

The matching block consisting of maskable CAM cells compares the window pattern to all templates stored in the content-addressable memories in parallel, and N maskable CAM cells are tied together to a match line where N is the number of bits in a window pattern. Because there is a match line for each template, the match signal is the output of an OR operation on all match lines. A maskable CAM cell with 17 transistors (Figure 8) can store one bit for the pattern word and one for the mask word. Two custom ICs have been designed and are being fabricated. A RED IC can store 128 templates for the 25-bit window and a DED IC can contain 64 templates for the 32-bit window.

Figure 9 is a block diagram of a prototype inspection system. The input is a serial binary image of the VLSI circuit patterns, and the output is a color display. The delays are used to locate the centers of the RED and DED windows at the same pixel and to synchronize the outputs of the RED and DED chips to the delayed input signal, which corresponds to the center pixel of the RED and DED windows.

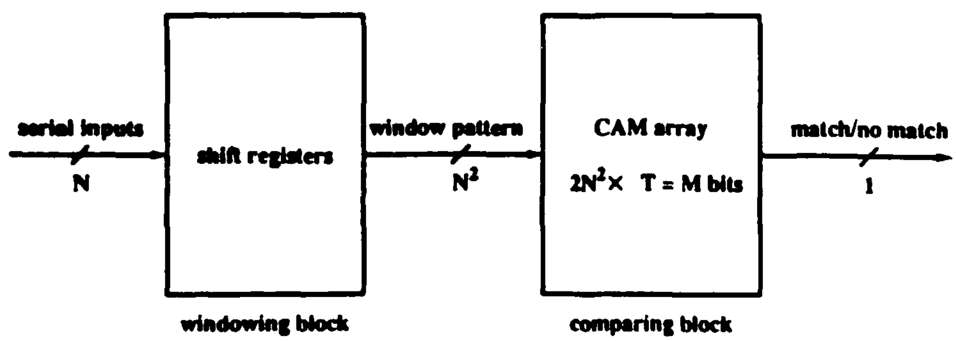


Figure 7. Functional diagram of an VLSI inspection circuit.

Conclusion

The template-set approach is a simple and effective method for resolving the problem of VLSI circuit pattern inspection. A real-time inspection can be accomplished with fast content-addressable memories. The advantages of this approach are summarized as follows.

- "Don't-care" is employed to reduce the number of templates and to resolve the problem of quantization error effectively.
- The maskable content-addressable memory is used as a storage for the template set and parallel-comparison unit.
- The content-addressable memory required to store the template sets is orders of magnitudes smaller in volume than the CAD data of the patterns.

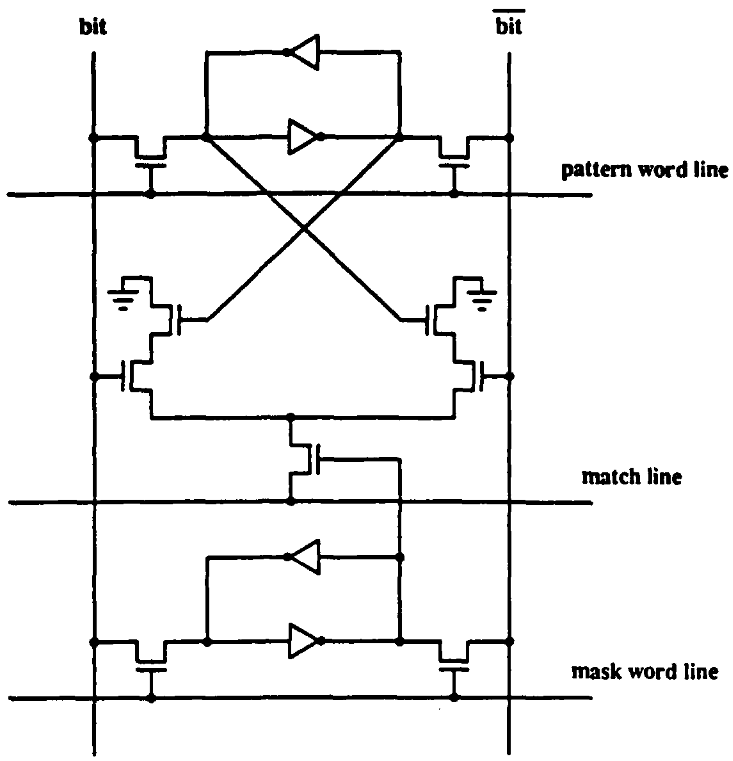


Figure 8. Circuit diagram of a maskable content-addressable memory cell.

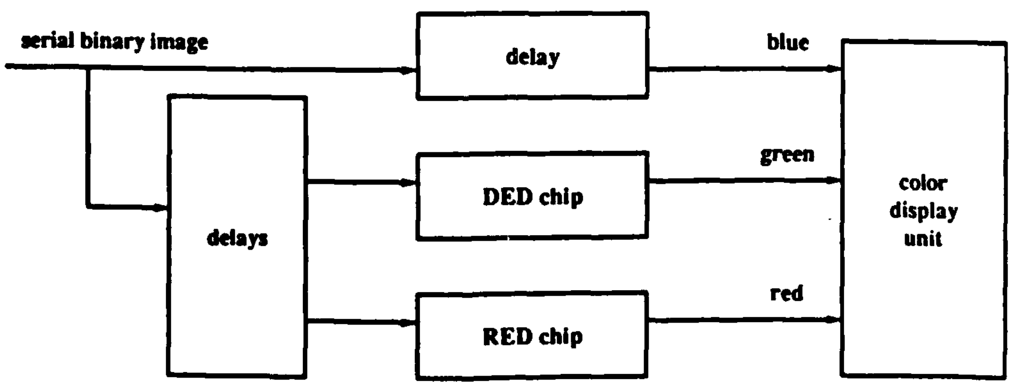


Figure 9. Block diagram of a prototype inspection system.

- The template set can be adjusted if the set of error patterns to be covered is changed.
- Multiple template sets can be employed to enhance defect coverage if a template set cannot detect all the error patterns.
- Real-time inspection is possible because of the high speed and parallelism of a CAM array.
- This approach is compatible with the raster-scanned image because the window pattern at every pixel position must be compared to the template set.

Because outputs of this inspection method are positions where the unacceptable window patterns exist, it can be used for analyzing the unacceptable window patterns to classify the type of defects, which is an area of further research.

Acknowledgment

The authors are grateful to Dr. P. A. Crossley, a CIS visitor from the Materials Analysis Laboratory of Fairchild Research, Palo Alto, California, for the SEM image data. This work was supported jointly by the Semiconductor Research Corporation under Contract SRC-84-01-046 and Defense Advanced Research Projects Agency under Contract MDA903-84-K-0062.

References

1. R. T. Chin and C. A. Harlow, "Automated Visual Inspection : A Survey", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Nov, 1982, pp. 557-573.

2. M. Ejiri, T. Uno, M. Mese, and S. Ikeda, "A Process for Detecting Defects in Complicated Patterns", *Computer Graphics and Image Processing*, 1980.

3. N. Goto and T. Kondo, " An Automatic Inspection System for Mask Pattern", *Proc. 4th Intl. Joint Confon. Pattern Recognition*, 1978.

4. J. F. Jarvis, "A Method for Automating the Visual Inspection of Printed Wiring Boards", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Jan, 1973, pp. 77-82.

5. C. Mead and L. Conway, *An Introduction to VLSI Systems*, Addison Wesley, 1980.

END

8-87

DTIC